

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practise in the
Company

2010

Marek Janáček

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. května 2010

.....
podpis

Poděkování

Tímto bych chtěl poděkovat Ing. Kristianu Wiglaszovi za hodnotné rady a odborné vedení během mé odborné praxe.

Abstrakt

Tato bakalářská práce se zabývá průběhem odborné praxe ve firmě Tieto. V první části je představena firma Tieto a pracovní zařazení studenta v rámci odborné praxe. Dále jsou uvedeny a popsány úkoly studentovi během praxe zadané. U úkolu jsou potom zmíněny, jaké postupy a technologie při jejich řešení student použil. V předposlední části jsou shrnuty znalosti, které student během praxe uplatnil, a také ty, které postrádal. V závěru se práce věnuje zhodnocení výsledků odborné praxe a jejímu celkovému hodnocení.

Klíčová slova

.NET, C#, XML, WinForms, Doxygen, Tieto, Windows Search, VBA, dokumentace

Abstract

This thesis deals with professional practise in the company Tieto. The first part is a presentation of the company Tieto and student`s grade during the professional practise. Next there are listed and described tasks which student work on. For each task there are mentioned technologies and methods used by student. The penultimate section summarizes the knowledge that the student has applied during practise, and those who lacked. In conclusion the thesis presents the results of evaluation of professional practise and its overall rating.

Keywords

.NET, C#, XML, WinForms, Doxygen, Tieto, Windows Search, VBA, documentation

Seznam použitých symbolů a zkratek

API - Application Programming Interface

ERP - Enterprise Resource Planning

MSDN - Microsoft Developer Network

SQL - Structured Query Language

SŘBD - Systém řízení báze dat

VBA - Visual Basic for Applications

XML - Extensible Markup Language

Obsah

1	Úvod.....	1
2	Firma Tieto a pracovní zařazení.....	2
2.1	Tieto	2
2.2	Pracovní zařazení	2
3	Úkoly zadané během odborné praxe	3
3.1	Generování programátorské dokumentace k systému Fenix	3
3.2	Nástroj pro vytváření testovacích zpráv na základě definovaných šablon.....	3
3.3	Aplikace pro vizualizaci obsahu zpráv během ladění	4
3.4	Automatizace procesu sestavování výsledků zátěžových testů.....	5
3.5	Analýza a vývoj nadstavby pro Windows Search	5
4	Zvolený postup řešení zadaných úkolů	7
4.1	Generování programátorské dokumentace k systému Fenix	7
4.2	Nástroj pro vytváření testovacích zpráv na základě definovaných šablon.....	7
4.3	Aplikace pro vizualizaci obsahu zpráv během ladění	9
4.4	Automatizace procesu sestavování výsledků zátěžových testů.....	9
4.5	Nadstavba pro Windows Search	9
5	Znalosti a dovednosti získané v průběhu studia uplatněné v průběhu odborné praxe	11
6	Znalosti a dovednosti scházející studentovi v průběhu odborné praxe	11
7	Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení	12
	Seznam literatury	13
	Seznam obrázků	14
A	Obsah CD.....	15

1 Úvod

Jedním z důležitých částí výuky během studia je praxe. Během praxe může student uplatnit a použít teoretické znalosti a dovednosti nabyté během studia. Rovněž má možnost čelit reálným problémům, a tím se blíže seznámit se skutečnými postupy používanými v praxi.

V této bakalářské práci nejprve představím firmu, ve které jsem odbornou praxi vykonával a pozici, na které jsem pracoval. V další kapitole uvedu a stručně popíšu úkoly, které mi byly zadány. U jednotlivých úkolů se pokusím nastínit, jak jsem je řešil, jaké technologie jsem při jejich řešení používal, a s jakými problémy jsem se musel potýkat. V předposlední části se budu zabývat, které znalosti získané během studia jsem během praxe uplatnil a které mi naopak scházely. V úplném závěru potom zhodnotím dosažené výsledky a praxi jako celek.

2 Firma Tieto a pracovní zařazení

2.1 Tieto

Společnost Tieto v dnešní podobě vznikla spojením finské společnosti Tieto, založené v roce 1963 a švédské společnosti Enator, založené v roce 1995, které se uskutečnilo v roce 1999. Do roku 2009 pak vystupovala pod jménem TietoEnator, kdy se její jméno zkrátilo na Tieto.

Tieto působí na trhu poskytování služeb v oblasti informačních technologií, výzkumu, vývoji a poradenství. Hlavními trhy Tieto jsou severní Evropa, Rusko a Německo, kde poskytuje své služby především středně velkým a velkým organizacím. Hlavními obchodními partnery jsou především organizace z oblastí telekomunikací, digitálních služeb, bankovníctví, zdravotnictví, lesního, ropného a plynárenského průmyslu.

V současné době Tieto působí téměř ve 30 zemích světa a zaměstnává okolo 16000 pracovníků. V České republice působí od roku 1995. Hlavním místem působnosti v České republice je Ostrava, kde se také nalézá Czech Global Delivery Centre, ve kterém nyní pracuje okolo 1500 zaměstnanců a patří mezi významné světové pobočky. Tieto je rovněž nejvýznamnější společnost podnikající v oblasti informačních technologií v Moravskoslezském kraji.

2.2 Pracovní zařazení

V rámci své odborné praxe jsem byl zařazen do ostravského oddělení Microsoft Service Practice a přidělen na projekt Fenix na pozici Software Specialist. Fenix je rozsáhlý ERP systém vyvinutý v 90. letech výhradně pro finskou společnost StoraEnso, zabývající se dřevozpracujícím a papírenským průmyslem. V současné době je systém nasazen v provozu a pracuje se na odstraňování chyb a implementaci nových požadavků od zákazníka. Nicméně i přesto pracují na tomto projektu stále desítky lidí.

Náplň mé práce během odborné praxe nebyla přímo implementace, či testování Fenixu, ale i přesto jsem měl možnost úzce spolupracovat s řadou vývojářů a testerů, a díky toho blíže poznat jejich každodenní práci. Stěžejními oblastmi mé práce bylo generování programátorské dokumentace k systému Fenix, a také vývoj nástrojů, které by zefektivnily práci vývojářům a testerům. To samozřejmě zahrnovalo přímou komunikaci se spolupracovníky, abych poznal a pochopil, jak jejich práce probíhá a jaké jsou jejich představy na její zefektivnění. Nakonec bylo vždy nutné k jednotlivým nástrojům vytvořit přehlednou dokumentaci, případně je i kolegům předvést tak, aby je mohli začít snadno používat a plnily svůj účel.

3 Úkoly zadané během odborné praxe

3.1 Generování programátorské dokumentace k systému Fenix

Prvním z mých hlavních úkolů během odborné praxe bylo generování programátorské dokumentace k systému Fenix. Dokumentace se vytvářela pomocí nástroje Doxygen [1]. Jelikož se stále na Fenixu pracuje, je nutné v určitých časových intervalech vygenerovat z aktuálních zdrojových kódů novou dokumentaci tak, aby byla její aktuální verze k dispozici všem vývojářům na intranetu. Jak jsem již uvedl, je systém Fenix velice rozsáhlá aplikace a v tom tkvěl největší problém tohoto úkolu. Zdrojové kódy Fenixu mají v současné době několik miliónů řádků kódu a při procesu generování se naráželo na problém limitu velikosti alokované paměti 32 bitových aplikací na platformě Windows. Tomuto úkolu jsem se věnoval několik pracovních dní především na začátku své praxe. V jejím dalším průběhu jsem se mu pak věnoval jen příležitostně.

3.2 Nástroj pro vytváření testovacích zpráv na základě definovaných šablon

Mým druhým úkolem bylo vytvořit nástroj, který by usnadnil práci testerům a vývojářům při testování systému Fenix v oblasti zasílání zpráv mezi klientem a serverem. Práce při testování probíhala tak, že se musela ručně vytvářet testovací zpráva ze šablon ve formátu XML [2]. Proces vytváření zprávy vypadal tak, že bylo nutné hledat v čistě textové šabloně příslušná místa pro vyplnění konkrétních hodnot, kopírovat, či mazat některé opakující se části podle potřeby, vkládat počty, či indexy těchto částí, nebo vkládat aktuální datum. Samotné vytváření zprávy bylo proto poněkud nepohodlné a také náchylné k chybám. Nakonec byla vytvořena testovací zpráva odeslána na server.

Úkolem pro můj nástroj bylo umět vytvářet tyto zprávy na základě šablon pro uživatele pohodlnějším a hlavně rychlejším způsobem. Šablona se obecně skládala z částí, která byla vždy stejná a pak ze speciálních částí, které byly vyplňovány na základě vstupu od uživatele, nebo i zcela automaticky. Pro označení částí šablon vznikl velice jednoduchý značkovací jazyk. V šabloně bylo možno definovat sekce, což byly části, které se mohly ve výsledné zprávě libovolně-krát opakovat dle potřeby. Tyto sekce byly identifikovatelné jménem, bylo možné se ptát na jejich počet a index, to znamená zapsat tuto informaci do výsledné zprávy na definované místo. Sekce bylo možné i neomezeně zanořovat. Dále bylo možné definovat proměnné, jejich výchozí hodnoty a místa, kam se ve výsledné zprávě zapíší. Hodnoty těchto proměnných byly získávány na základě vstupu od uživatele. Proměnné mohly být globální, či lokální v rámci sekce. V rámci jedné šablony bylo možné vkládat do jejího těla šablonu jinou. Rovněž šablona umožňovala vkládat na základě značek datum a čas v daném formátu a náhodné číslo

z příslušného intervalu. Jako poslední zajímavost bych zmínil možnost definovat aplikace, které budou po ukončení generování zprávy spuštěny s příslušnými parametry. Tato funkcionality byla užitečná především z důvodu možnosti automaticky spustit odesílání zprávy na server.

Uživatelské rozhraní dalo uživateli na výběr z množiny existujících šablon. Po výběru některé z nich byl zobrazen seznam sekcí, jejích instancí (ve výchozím stavu 1) a proměnných. V každé instanci sekce byl potom ještě seznam jejích lokálních (sekčních) proměnných a sekcí. U proměnných uživatel zadával jejich hodnoty, případně mohl ponechat vyplněnou hodnotu výchozí. U sekcí pak zadával počet jejích instancí. Po změně počtů instancí došlo samozřejmě ke změně zobrazovaných počtů instancí příslušné sekce, aby bylo možné zadat hodnoty její sekčních proměnných.

Jelikož se jednalo o poněkud komplikovanější aplikaci jak logikou, tak uživatelským rozhraním, zabrala samotná implementace této aplikace odhadem 14-16 dnů. Dalších několik dní si pak vyžádalo odstraňování chyb a řešení drobných nedorozumění při zadávání úkolů.

3.3 Aplikace pro vizualizaci obsahu zpráv během ladění

Další můj úkol v rámci odborné praxe byl opět věnován vývoji aplikace pro účely testování a ladění chyb Fenixu. Ve Fenixu je použito několik technologií třetích stran, přičemž některé z nich neumožňují spolupráci s Visual Studií během procesu ladění, což samotné ladění činilo méně komfortním pro vývojáře. Především šlo o absenci možnosti prohlížet obsah zpráv, zasílaných mezi klientem a Tuxedo [3] serverem. Již před mým příchodem na praxi byl vyvinut nástroj, který uměl obsah zprávy vypsat do textového souboru na disku. Tato funkcionality byla ale stále nedostačující, přesto tento nástroj našel uplatnění jako zdroj dat pro mou aplikaci.

Požadavky na mnou vyvíjenou aplikaci byly takové, aby fungovala podobně jako prohlížeč proměnných během ladění ve Visual Studiu. Nakonec ovšem byla funkcionality značně rozšířena. Aplikace zobrazovala obsah jednotlivých zpráv v záložkách, přičemž bylo možné si jednotlivé zprávy pro lepší orientaci přejmenovat, protože ve výchozím stavu byly názvy zpráv tvořeny hexadecimálními čísly. Toto pojmenování bylo perzistentní. Pro každou zprávu si aplikace od spuštění pamatovala všechny verze dané zprávy. Verzemi bylo možné listovat, dávat jim jména a také je mezi sebou libovolně porovnávat. Porovnávání bylo implementováno pomocí podbarvování změněných hodnot. Barvy si nastavoval uživatel dle potřeby. U každé verze zprávy aplikace ukazovala, kdy byla vytvořena, a bylo možné synchronizovat ostatní zprávy na verze, které vznikly v nejmenším časovém odstupu.

Na grafické rozhraní byly kladeny nároky související s provozem při ladění. Aplikace měla umožňovat měnit průhlednost hlavního okna, možnost umístit okno vždy na popředí, či jednoduše skrýt a znovu zobrazit části nástrojové lišty. Samozřejmostí byla možnost hledání v obsahu zobrazené zprávy, či kopírování skrze schránku operačního systému.

Tato aplikace byla vytvářena ve dvou verzích. Prvně jako samostatně stojící aplikace, ale také jako plugin pro Microsoft Visual Studio ve verzích 2005 a 2008. Drtivá většina kódu se i tak podařila zachovat identická pro obě verze. Nicméně malé odchylky se přece jenom nakonec vyskytly. Například není možné měnit průhlednost u pluginu.

Práce na této úloze zabrala zhruba 15 pracovních dnů. Po jejím dokončení bylo samozřejmě ještě potřeba několik dní na odladění chyb, aby mohla být aplikace nasazena k účelu, na který byla zamýšlena.

3.4 Automatizace procesu sestavování výsledků zátěžových testů

Obsahem tohoto úkolu bylo nejprve pochopit proces sestavování výsledků zátěžových testů, a poté se pokusit navrhnout, jak tento proces optimalizovat z hlediska časové náročnosti a především tuto optimalizaci a automatizaci pomoci realizovat.

Na systému Fenix kriticky závisí korektní fungování společnosti StoraEnso. Proto je nutné velice pozorně dbát na jeho pokud možno nulovou chybovost, ale také je nutné simulovat, jak se systém chová pod zátěží. Pro tento účel slouží zátěžové testy, na základě jejich výsledků je pak možno učinit opatření, aby nedošlo k nedostupnosti systému, například vlivem jeho přetížení.

Při zátěžovém testu se pomocí specializovaného nástroje měří zatížení SŘBD, délka jednotlivých typů transakcí a vytížení serverů, například Tuxedo nebo Windows serveru. Tato data jsou potom exportována do textových souborů a následně byla ručně tříděna a vkládána do předem připravených šablon v Excelu. Následně se vytváří porovnání s naměřenými daty z předchozích verzí systému a vytváří se sestavy, jak pro interní použití, tak i pro zákazníka. Tato ruční práce s Excelem byla poměrně časově náročná a mým úkolem bylo se ji co nejvíce pokusit automatizovat pomocí maker.

Vypracování tohoto úkolu i s odladěním mi zabralo 7 pracovních dnů.

3.5 Analýza a vývoj nadstavby pro Windows Search

Mým posledním úkolem bylo analyzovat možnosti, jak rozšířit nástroj Windows Search [4] a ve zbylém čase se pokusit toto rozšíření realizovat. Důvodem potřeby tohoto nástroje je obrovské množství zdrojových kódů systému Fenix, kdy vývojáři potřebují možnost fulltextového vyhledávání nad soubory se zdrojovými kódy, ale základní funkčnost Windows Search je pro ně nedostačující. Jejich potřebám především chybí možnost automatického zvýrazňování hledaných frází v živém náhledu zdrojového souboru.

Pro tento účel se dříve používala aplikace Copernic [5], která byla k užívání zdarma i pro firemní účely, licenční podmínky se ovšem změnily tak, že by Tieto muselo tuto aplikaci koupit, proto se raději zvolila možnost se pokusit aplikaci vyvinout vlastními silami přímo na míru.

Bohužel na realizaci tohoto projektu zbylo v rámci mé odborné praxe jen málo času, a proto se mi podařilo pouze danou oblast analyzovat a naprogramovat funkční prototyp této aplikace, na kterém by ovšem bylo ještě potřeba doimplementovat řadu funkcí, týkajících se především uživatelského rozhraní a rovněž jej otestovat, než by mohl být nasazen ke skutečnému užívání.

4 Zvolený postup řešení zadaných úkolů

4.1 Generování programátorské dokumentace k systému Fenix

Jak jsem již uvedl, pro generování dokumentace byl použit nástroj Doxygen a hlavním problémem byl limit velikosti adresovatelné paměti u 32 bitových aplikací na platformě Windows na 2GB pro jeden proces. Při generování dokumentace z kompletních zdrojových kódů docházelo vždy k pádu Doxygenu z důvodu vyčerpání dostupné paměti. Vzhledem k množství zdrojových kódů trvalo generování kompletní dokumentace okolo 12 hodin.

Částečným řešením se nakonec ukázalo vyloučení části zdrojových souborů z procesu generování. Takovéto řešení bylo ale poněkud nepraktické, protože část systému v takovémto případě nebyla dokumentována.

Dále jsem zkoušel různé verze Doxygenu, zda některá nebude paměťově úspornější, to ale bez úspěchu. Tento problém jsem také konzultoval na oficiálním fóru věnujícímu se vývoji a odstraňování chyb Doxygenu. Odpovědí bylo, že se skutečně jedná o limit 32 bitové technologie. 64 bitová verze Doxygenu k dispozici nebyla.

Po řadě pokusů se nakonec dobrým řešením ukázala kombinace následujících kroků

- Úprava souboru boot.ini, kdy byly přidány parametry “/3GB /USERVA=xxx”, kde xxx odpovídá maximální velikosti paměti přidělené jednomu procesu.
- Úprava spouštěcího souboru Doxygenu tak, aby uměl adresovat více než 2GB paměti pomocí nástroje editbin [6] s parametrem “/LARGEADDRESSAWARE”.

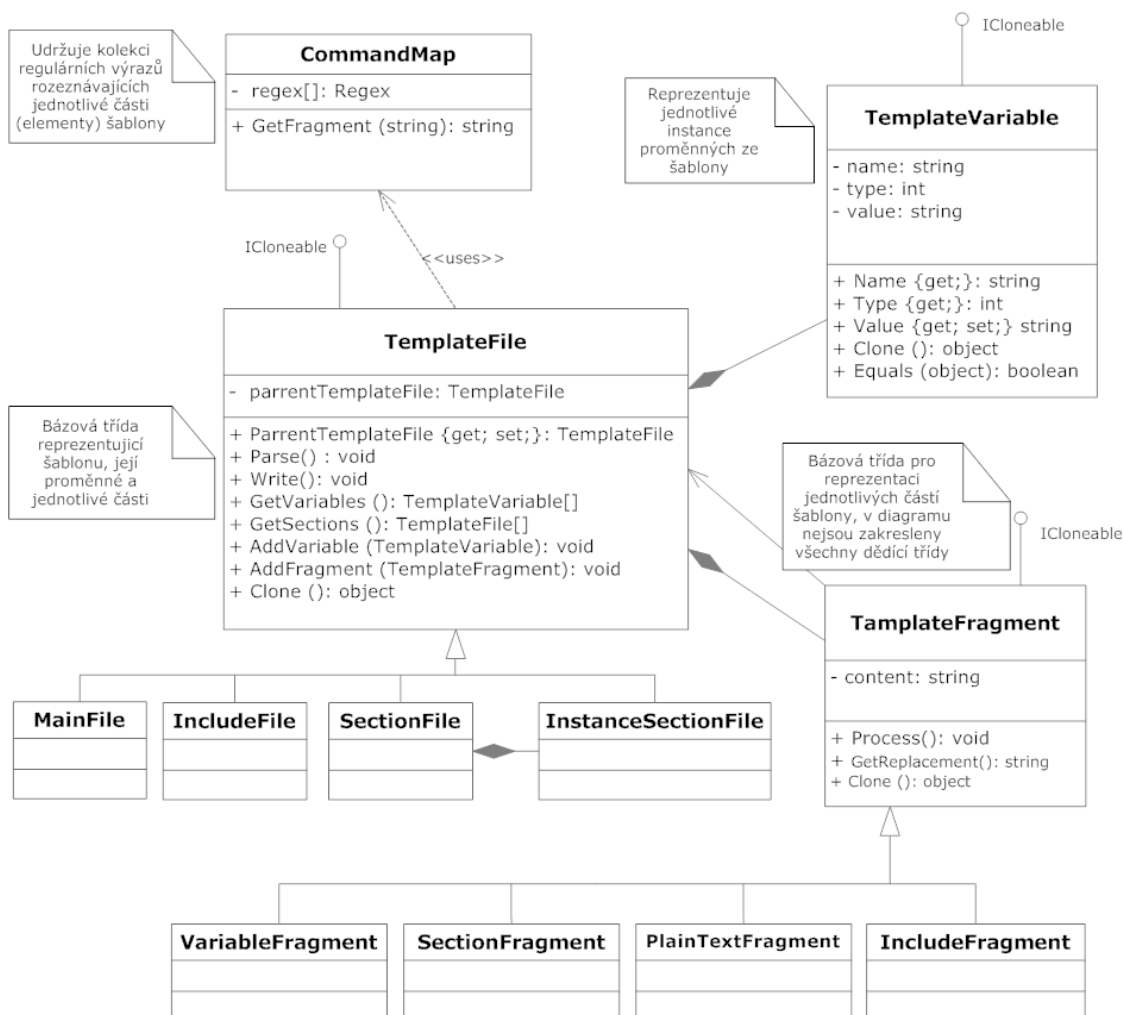
Tato kombinace kroků byla ovšem funkční jen s verzemi Doxygenu 1.5.5 a starších. U novějších verzí stále docházelo k pádům v důsledku nedostatku paměti.

4.2 Nástroj pro vytváření testovacích zpráv na základě definovaných šablon

Tento nástroj jsem implementoval v prostředí .NET [7], programovacím jazyce C# [8] a pro tvorbu uživatelského rozhraní jsem použil technologii WinForms [9]. Nejzajímavějšími a zároveň nejnáročnějšími oblastmi při vývoji byla samotná aplikační logika a současně také nutnost vyhledat, upravit a naučit se pracovat s grafickou komponentou, která není standardní součástí WinForms. Tuto komponentu bylo nutné využít z důvodu nároků kladených na uživatelské rozhraní.

Samotná aplikační logika zahrnovala vytvořit parser pro definovaný značkovací jazyk, což by samo o sobě bylo velice jednoduché. Méně snadné už ale bylo vytvořit objektový model šablony, který bylo následně, dle údajů zadaných uživatelem, možné měnit a z něj pak

generovat výstupní zprávu do formátu XML, to bylo nutné pro splnění požadavků na funkcionalitu. Samotná šablona ve formátu XML nebyla, čili použití existujících parserů možné nebylo.



Obrázek 1: Třídní diagram hlavní části aplikace

Nejnáročnější byla tvorba uživatelského rozhraní a to z důvodu potřeby grafické komponenty, který není součástí WinForms. Pro účely této aplikace byla potřeba komponenta kombinující funkcionalitu prvků Listview a Treeview. Pro potřeby mé aplikace jsem dostal za úkol najít a použít existující komponentu třetí strany takovou, na kterou by se nemusela kupovat licence. Nakonec jsem použil volně dostupnou komponentu s názvem TreeListView [10], která téměř splňovala požadavky na funkcionalitu, ale i tak ji bylo nakonec potřeba mírně přizpůsobit potřebám mé aplikace. Kvůli nedostatku dokumentace jsem nejprve byl nucen porozumět kódu, a to byl problém především z důvodu, že tato komponenta používala volání Windows API. Zpočátku moje práce probíhala systémem pokus omyl, nakonec jsem princip fungování

komponenty TreeListView pochopil a byl s ní schopen pracovat. Dále bylo nutné odstranit některé drobné chyby, které následně způsobovaly pád celé aplikace. Znalosti této komponenty jsem poté uplatnil i při vývoji další aplikace, kde také našla své uplatnění.

4.3 Aplikace pro vizualizaci obsahu zpráv během ladění

Tuto aplikaci jsem rovněž implementoval v prostředí .NET a programovacím jazyce C#, uživatelské rozhraní v technologii WinForms. Nejzajímavějším aspektem vývoje této aplikace byl z mého pohledu požadavek, aby mohla být provozována jako plugin pro Visual Studio. Při tvorbě uživatelského rozhraní opět našla uplatnění komponenta TreeListView z předešlého úkolu.

Během vývoje jsem se detailně seznámil s programováním formulářových aplikací na platformě .NET, a také jsem měl možnost nahlédnout na problematiku rozšiřování funkcionality samotného Visual Studia. Rovněž bylo zajímavé a přínosné získávat v průběhu vývoje zpětnou vazbu od mých spolupracovníků, pro které měla být tato aplikace určena. Jejich přání se totiž v průběhu vývoje poměrně měnila a stále měli snahu přidávat další funkcionalitu, která ani nebyla na začátku zamýšlena.

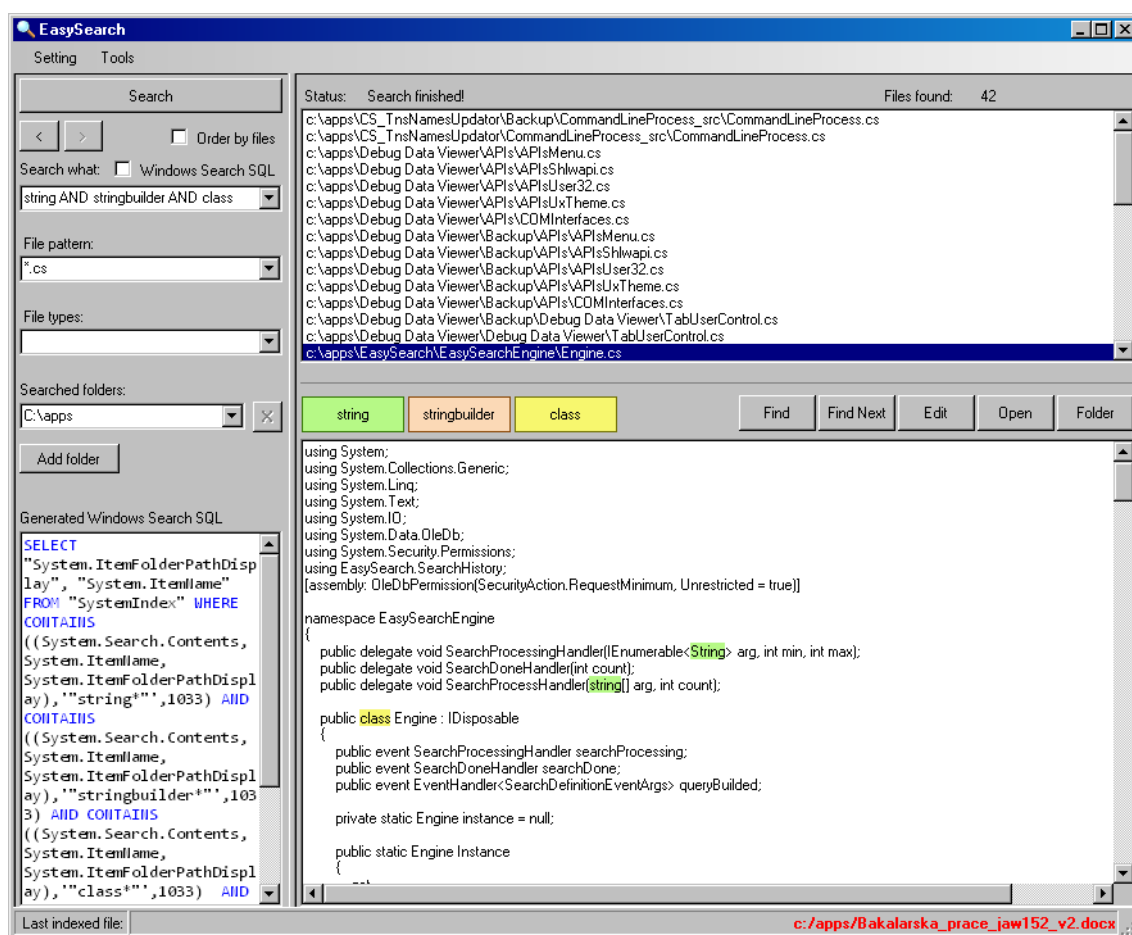
4.4 Automatizace procesu sestavování výsledků zátěžových testů

Zpočátku má práce probíhala tak, že mi byla vysvětlena problematika sestavování výsledků zátěžových testů a současně i poukázáno, které oblasti bych se měl pokusit zefektivnit. Poté přišla na řadu samotná implementace. Nejprve jsem se pokusil daný problém vyřešit pomocí relativně jednoduchého programu napsaného v prostředí .NET. Nicméně přání zadavatele bylo takové, aby si v budoucnu mohl sám zdrojové kódy dle potřeby jednoduše upravit. Proto mi bylo doporučeno vyřešit tento úkol pomocí maker napsaných ve VBA [11]. S programováním maker jsem neměl do té doby žádné zkušenosti, a tak mě nejprve čekalo rychlé samostudium, které mi usnadnilo několik již existujících maker, která byla na pracovišti vytvořena. Nakonec se mi zadaný problém podařilo vyřešit pomocí maker, a jelikož zadavatel byl s výsledky spokojen, rozhodli jsme se pokusit automatizovat i další části sestavování výsledků testů.

4.5 Nadstavba pro Windows Search

Na začátku řešení tohoto úkolu jsem se musel především seznámit s možnostmi Windows Search a prozkoumat možnosti, jak s tímto nástrojem spolupracovat. Hlavním zdrojem informací mi byly oficiální návody na MSDN [12], nicméně tyto návody nepokryly celou problematiku řešené úlohy a některé informace bylo nutno získávat ze specializovaných internetových fór. Za velice cenné bych označil zejména informace z webu The Code Project [13].

V druhé fázi mé práce na tomto úkolu mě čekala samotná implementace. Aplikace byla opět postavena na platformě .NET. Stěžejním prvkem aplikace byl programový přístup k indexům Windows Search. Ten je uskutečněn pomocí velice zjednodušené verze jazyka SQL. Dále bylo nutné navrhnout a implementovat uživatelské rozhraní, které by splňovalo nároky mých spolupracovníků. Inspirací pro něj se stal nástroj Copernic.



Obrázek 2: Ukázka hlavního okna nadstavby pro Windows Search

Jak jsem již uvedl, aplikaci se z časových důvodů nepodařilo zcela dokončit. Především by bylo potřeba doimplementovat možnost ovlivnit, které adresáře a jaké typy souborů se budou indexovat. Další oblastí, kterou by bylo potřeba vylepšit je uživatelské rozhraní, zejména jeho funkce, které zdaleka nedosahuje komfortu, který je požadován a na který byli uživatelé zvyklí při používání aplikace Copernic.

5 Znalosti a dovednosti získané v průběhu studia uplatněné v průběhu odborné praxe

V průběhu odborné praxe jsem zužitkoval celou řadu znalostí a dovedností nabytých v rámci studia. Především bych chtěl zmínit obecný přehled z oblasti informačních technologií. Mezi nejcennější znalosti bych osobně zařadil ty získané v kurzech Základy algoritmizace, Programování v jazyce C#, Architektury počítačů, Úvod do softwarového inženýrství a Teorie zpracování dat. Našly se ale i předměty, jejichž znalosti uplatnění nenašly, především z oblastí elektroniky a telekomunikací.

Neméně důležitá byla i znalost anglického jazyka, jelikož firemním jazykem v Tietu je právě angličtina. Další cennou dovedností byla schopnost samostudia a rychlého zorientování se v aktuálně řešeném problému, kterou jsem se naučil během studia.

6 Znalosti a dovednosti scházející studentovi v průběhu odborné praxe

Jelikož jsem nastupoval na praxi jakožto student teprve začínajícího třetího ročníku bakalářského studia, chyběla mi celá řada znalostí a dovedností. Zejména bych zdůraznil menší praktickou zkušenost s vývojem reálných aplikací, kterou bohužel v rámci psaní semestrálních projektů získat asi nelze. Dále jsem postrádal lepší znalost návrhových vzorů a technologie .NET, proto jsem během praxe byl nucen hodně věcí nastudovat svépomocí. Bohužel předmět Architektura technologie .NET mě čekal až v 6. semestru. Dalšími oblastmi, kde jsem narážel na nedostatky znalostí, bylo programování s vlákny, a také mi chybělo alespoň základní povědomí o Windows API. Přestože jsem převážně pracoval s technologií .NET, občas by se, byť základní znalost Windows API, velice hodila.

Rovněž mi scházela například znalost pokročilých funkcí aplikace Excel, psaní maker ve Visual Basicu, práce s Doxygen, či verzovacím systémem Subversion [14]. To jsou ale natolik specifické oblasti znalostí, že mohou být jen stěží pokryty během tříletého studia.

7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Z mého pohledu byla odborná praxe pro můj budoucí profesní život velice přínosná. Práce na zadaných úkolech mě nutila se průběžně učit řadu nových zajímavých věcí. Z odborných dovedností, získaných během praxe, bych zdůraznil především získání zkušeností s vývojem reálných aplikací. Dále práci z verzovacím nástrojem Subversion a generátorem programátorské dokumentace Doxygen. Zároveň bych ale chtěl zmínit i neméně důležitou oblast a to, že jsem měl možnost poznat, jak funguje společnost zabývající se informačními technologiemi velikosti Tieta, jak vypadá práce na skutečném softwarovém projektu. Rovněž bylo přínosné vyzkoušet si, jak důležitá je a jak vypadá komunikace mezi vývojářem v mé osobě a zadavatelem, či uživateli mých produktů v roli mých spolupracovníků a jak důležitá je i komunikace mezi samotnými vývojáři.

Dosažené výsledky za dobu mé odborné praxe hodnotím kladně. Z mého pohledu se mi dařilo úspěšně plnit zadané úkoly. Zároveň mě velice těší, že mnou vyvinuté aplikace našly uplatnění u mých spolupracovníků a že jsou jimi používány.

Seznam literatury

- [1] Doxygen. [Online] Získáno 3 2010, z www.doxygen.org/.
- [2] W3C. [Online] Získáno 4 2010, z <http://www.w3.org/XML/>.
- [3] Oracle Corporation *Oracle Tuxedo*. [Online] Získáno 3 2010, z <http://www.oracle.com/technology/products/tuxedo/index.html>.
- [4] Microsoft Corporation *Windows Search Overview*. [Online] Získáno 4 2010, z [http://msdn.microsoft.com/en-us/library/aa965362\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/aa965362(VS.85).aspx).
- [5] Copernic *Copernic Desktop Search*. [Online] Získáno 3 2010, z <http://www.copernic.com/en/products/desktop-search/index.html>.
- [6] Microsoft Corporation *Editbin Reference*. [Online] Získáno 4 2010, z <http://msdn.microsoft.com/en-us/library/xd3shwhf.aspx>.
- [7] Microsoft Corporation *.NET Development*. [Online] Získáno 4 2010, z [http://msdn.microsoft.com/cs-cz/library/ff361664\(en-us\).aspx](http://msdn.microsoft.com/cs-cz/library/ff361664(en-us).aspx).
- [8] Microsoft Corporation *MSDN Visual C# Developer Center*. [Online] Získáno 4 2010, z <http://msdn.microsoft.com/en-us/library/kx37x362.aspx>.
- [9] Microsoft Corporation *Windows Forms Overview*. [Online] Získáno 4 2010, z [http://msdn.microsoft.com/en-us/library/8bxxxy49h\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/8bxxxy49h(VS.80).aspx).
- [10] The Code Project *TreeView*. [Online] Získáno 4 2010, z <http://www.codeproject.com/KB/list/treelistview.aspx>.
- [11] Microsoft Corporation *Visual Basic Developer Center*. [Online] Získáno 4 2010, z <http://msdn.microsoft.com/en-us/vbasic/default.aspx>.
- [12] Microsoft Corporation *MSDN*. [Online] Získáno 4 2010, z <http://msdn.microsoft.com/cs-cz/library/default.aspx>.
- [13] The Code Project. [Online] Získáno 4 2010, z <http://www.codeproject.com/>.
- [14] Tigris.org *TortoiseSVN*. [Online] Získáno 4 2010, z <http://tortoisesvn.tigris.org/>.

Seznam obrázků

1	Třídní diagram hlavní části aplikace	8
2	Ukázka hlavního okna nadstavby pro Windows Search	10

A Obsah CD

Adresářová struktura přiloženého CD

Adresář	Popis
\doc	Elektronická verze bakalářské práce
\src\DebugDataViewer	Zdrojové soubory aplikace pro vizualizaci zpráv
\src\EasySearch	Zdrojové soubory nadstavby pro Windows Search
\src\LoadTesting	Soubory obsahující makra pro automatizaci sestavování výsledků zátěžových testů
\src\XmlMsgEdit	Zdrojové soubory aplikace pro vytváření testovacích zpráv